

## TUSTEP im WWW-Zeitalter: Werkzeug für Anwender und Programmierer

Kopp, Matthias; Küster, Marc Wilhelm; Ott, Wilhelm

Veröffentlichungsversion / Published Version  
Zeitschriftenartikel / journal article

Zur Verfügung gestellt in Kooperation mit / provided in cooperation with:  
GESIS - Leibniz-Institut für Sozialwissenschaften

### Empfohlene Zitierung / Suggested Citation:

Kopp, M., Küster, M. W., & Ott, W. (2000). TUSTEP im WWW-Zeitalter: Werkzeug für Anwender und Programmierer. *Historical Social Research*, 25(1), 143-151. <https://doi.org/10.12759/hsr.25.2000.1.143-151>

### Nutzungsbedingungen:

Dieser Text wird unter einer CC BY Lizenz (Namensnennung) zur Verfügung gestellt. Nähere Auskünfte zu den CC-Lizenzen finden Sie hier:  
<https://creativecommons.org/licenses/by/4.0/deed.de>

### Terms of use:

This document is made available under a CC BY Licence (Attribution). For more Information see:  
<https://creativecommons.org/licenses/by/4.0>

## TUSTEP im WWW Zeitalter: Werkzeug für Anwender und Programmierer

*Matthias Kopp, Mare Wilhelm Küster, Wilhelm Ott (Tübingen)\**

Das Thema fragt nach der Rolle eines lange bewährten Werkzeugs in einer veränderten Umwelt. Der Untertitel weist ihm eine immer kleiner werdende Nische im großen Feld der Computer-Anwendungen zu und benennt gleichzeitig das Teilgebiet, auf dem wir glauben, Maßstäbe gesetzt zu haben und auf das wir uns weiterhin konzentrieren wollen.

In den 25 Jahren, in denen wir uns mittlerweile in diesem Kreis treffen, hat der Computer einen grundlegenden Wandel durchgemacht: er ist vom Werkzeug zum Medium geworden. Als Werkzeug existiert der Computer zwar weiter und wird immer leistungsfähiger und unverzichtbarer – dennoch ist er im allgemeinen Bewußtsein in dieser Rolle kaum mehr präsent; und selbst im wissenschaftlichen Bereich spielt er (außer vielleicht als Ersatz für die Schreibmaschine) seine Rolle überwiegend als Informations- und Kommunikationsmedium.

Um das »öffentliche Bewußtsein« zu charakterisieren, möchte ich die jüngste Marktstudie »Junge Zielgruppen« des Ehapa-Verlags zitieren. Danach spielt der Computer in diesem Konsum-Umfeld eine durchaus bedeutende Rolle – eingeordnet in die Rubriken »Unterhaltungselektronik« oder »Medienausstattung«. Man liest dort, daß in 46% der Haushalte, in denen Kinder unter 15 Jahren leben, ein Computer vorhanden ist, gegenüber in nur 23% der Haushalte ohne Kinder. 24% aller 6- bis 17jährigen spielen mit ihrem PC. Im Durchschnitt haben die Eltern 2.679,- DM für die Computerausstattung ausgegeben. Bei den Assoziationen, die Jugendliche mit dem Computer verbinden, kommt so etwas wie »Werkzeug« oder »Arbeitsmittel« gar nicht vor; in der Diskussion sind vielmehr Begriffe wie »Sicherheit der Privatsphäre«, »Isolation des Einzelnen« bzw. »Bedeutung von Multimedia« und »Verbesserung der Lebensqualität«.

Zurück zur Universität. Auch hier stehen längst die Informations- und Kommunikationsdienste im Zentrum, wie z.B. das Faltblatt zeigt, mit dem sich das ZDV seit Mitte Oktober seinen Nutzern vorstellt: es »betreibt das Netzwerk und die zentralen Datenverarbeitungssysteme«; die Zahl der ans Uni-Netz und darüber ans Internet angeschlossenen Rechner, der tägliche Mail- und News-Durchsatz sowie die Zugriffe auf WWW-Seiten stehen im Vordergrund; ein Hinweis auf Werkzeuge für die Forschung und auf deren Leistung fehlt völlig;

---

\* Protokoll des 74. Kolloquiums über die Anwendung der EDV in den Geisteswissenschaften an der Universität Tübingen am 5. Dezember 1998.

dagegen wird dem WWW und Hinweisen zu seiner Nutzung eine von insgesamt zwei Textseiten gewidmet.

Im Bereich des »computing in the humanities« schaut es nicht anders aus. Seit einigen Jahren entstehen hier immer wieder Initiativen, die die Entwicklung von neuer Software zur Textanalyse vorantreiben wollen.

Eingeleitet hat dies Nancy Ide vom Vassar College im Juni 1993 mit der Ankündigung der »Text Software Initiative« (TSI, in Anlehnung an die Text Encoding Initiative, TEI, die sie 6 Jahre zuvor initiiert hatte). Ziel sollte sein, »to establish and publish guidelines and standards for the development of text Software« und »promulgate and coordinate the development of free TSI-conformant software.« (TEI-L vom 4. Juni 1993).

Im Frühjahr 1996 folgte das »Text Analysis Software Planning Meeting« in Princeton, über das ich in unserem Kolloquium vom Juli 1996 berichtet habe. Eine Beziehung zur TSI gab es nicht.

Eine dritte Initiative zur Text Analysis Software wurde auf der diesjährigen Konferenz der ACH/ALLC in Debrecen von Harold Short und John Bradley aus dem Londoner King's College gestartet.

Aus dem Treffen in Debrecen ist inzwischen die Elta Software Initiative hervorgegangen, initiiert von Tom Horton (Florida Atlantic University) und John Bradley (King's College London). Elta steht für »Encoded literary text analysis«. Ihr Ziel ist, »to encourage and support the development of Software to support the analysis and study of literary and linguistic texts that have been encoded using modern markup-Systems (e.g. SGML)« (aus der »Charta of Elta Software Initiative« vom 28. September 1998: <http://www.kcl.ac.uk/humanities/cch/elta>). Sie bleibt damit hinter einem Aufruf zurück, den John Dawson (Cambridge) und Bruce Robertson (U of Toronto) über die Humanist Discussion Group an John Bradley and Geoffrey Rockwell nach dem Treffen von Debrecen gerichtet hatten, nämlich »to invite ... colleagues to form a Text Handling Software Working Group (THSWG)«. Der Aufruf betraf »Text Handling Software«, die Elta beschränkt sich (wie schon der Titel der Diskussion in Debrecen) auf »Text Analysis« und schränkt diese ein auf Texte, die mit modernem Markup versehen sind.

Die bisher letzte Initiative ist ein »TA Software Meeting« in Bergen vom 23. und 24. Oktober dieses Jahres, hervorgegangen aus Diskussionen zwischen Manfred Thaller (Bergen, Humanistisk Datasenter) und Harold Short (King's College, London). Von den in Elta engagierten Personen hat John Bradley teilgenommen.

Allen diesen Initiativen gemeinsam ist, daß sie sich um Software zur Textanalyse bemühen wollen – also um den Zugriff auf und die Auswertung von fertigen, mit Markup versehenen Texten, die z. B. im Netz verfügbar sind. Es geht also nicht um Hilfsmittel für Erstellung, Kontrolle, Erschließung und Bereitstellung von Texten, sondern für die Nutzung von durch andere bereitgestellten und aufbereiteten Texten.

Daraus ergeben sich auch die Forderungen, die an solche Software gestellt werden: »To prove really 'useful', these tools will have to be easy enough for the non-hacker to use effectively (which basically means, to play with using texts 'off the shelf )« (Wendell Piez, Elta mailing list vom 15. 10. 1998). Der Architekturvorschlag von John Bradley schließt – als ersten Punkt – sogar »a collection of textual objects« mit ein.

Im Mittelpunkt dieser Diskussionen steht also, sehr plakativ ausgedrückt, der (wissenschaftliche) Konsum von elektronisch verfügbaren Texten; ihr Vorhandensein wird vorausgesetzt.

Zurück zu unserem Thema »TUSTEP im WWW-Zeitalter«. TUSTEP tritt mit dem Anspruch auf, beide Gebiete, nämlich Produktion und wissenschaftliche Nutzung, abzudecken. Wir bezeichnen es als »leistungsfähiges Werkzeug zum wissenschaftlichen Umgang mit Textdaten ... vor allem für diejenigen Wissenschaften, in denen Texte Objekte der Forschung sind: ... Wissenschaften also, in denen nicht nur neue Texte als Produkt der eigenen wissenschaftlichen Arbeit erstellt und publiziert werden, sondern in denen schon existierende, überlieferte, schriftlich fixierte oder zu fixierende Texte ... durch kritische Neuedition gesichert, sprachlich und stilistisch analysiert, inhaltlich erschlossen, bibliographisch erfaßt werden müssen«.

Dem tragen die Design-Prinzipien Rechnung, die ich – wie schon u. a. auf dem bereits erwähnten Princeton Meeting – mit vier Schlagworten zusammenfassen möchte:

- Modularität: als Voraussetzung für die erforderliche Flexibilität: TUSTEP besteht aus voneinander relativ unabhängigen Programmen. Jedes stellt nur bestimmte Grundoperationen der Textdatenverarbeitung zur Verfügung und kann dabei über Parameter den Bedürfnissen der jeweiligen Aufgabenstellung angepaßt werden. Die einzelnen Module können zu Prozeduren zusammengestellt werden, die komplexe Aufgabenstellungen erledigen.
- Professionalität: Die verfügbaren Funktionen müssen auf den Bedarf ernsthafter Forschungstätigkeit abgestimmt sein und deshalb über die von OfficePaketen gebotenen Leistungen hinausgehen. Auf der System-Seite steht Professionalität für Eigenschaften wie Effizienz, Sicherheit und für die Fähigkeit, mit großen Datenmengen umzugehen.
- Integration: der einzelnen Module in ein System von aufeinander abgestimmten und beliebig kombinierbaren Programmen: Zusammen müssen sie alle Arbeitsphasen eines Projekts abdecken, von der Erfassung über die Analyse und Verarbeitung bis zur Ausgabe in konventioneller oder elektronischer Form.
- Portabilität: von Programmen und von Daten über verschiedene ComputerSysteme und über lange Zeiträume hinweg: Sie erst ermöglicht die Durchführung größerer Projekte trotz zwischenzeitlicher Generationswechsel bei EDV-Hardware und Betriebssystemen.

Dieses Konzept trägt zwei gegensätzlichen Anforderungen Rechnung: einerseits muß der Anwender die Möglichkeit (oder besser: gar keine andere Wahl) haben, als jeden einzelnen Schritt seines Lösungswegs selbst zu definieren; nur so weiß er wirklich, was ein Programm tut, und nur so kann er die Verantwortung für die Ergebnisse übernehmen. Andererseits müssen die Elementarschritte der Textdatenverarbeitung die vielen technischen Details von Hardware, Betriebssystemen und Programmiersprachen vor dem Anwender verbergen. Nur so bleibt der Lösungsweg überschaubar und beherrschbar.

Michael Sperberg-McQueen hat dieses Konzept in seinem Bericht über das Treffen in Princeton als »perhaps the most important points for thinking about the next generation of text analysis tools« bezeichnet. Seither haben wir weitere Anerkennung dafür bekommen – freilich jeweils verbunden mit dem Hinweis, daß die Benutzeroberfläche nicht mehr akzeptabel sei.

So war John Bradley nach seinem eigenen Bericht bei dem Treffen in Bergen (cf. [http://www.cse.fau.edu/\(ti\)tom/lists/elta-1/msg00008.html](http://www.cse.fau.edu/(ti)tom/lists/elta-1/msg00008.html)) der Ansicht, »that we could take as our starting points some of the transformational tools in TuStep«; und Willard McCarty schrieb mir in einem relativ ausführlichen Bericht über die Text Analysis Software Sitzung in Debrecen vom Sommer dieses Jahres folgende Zusammenfassung seiner Eindrücke:

»We are likely to remain in a murky situation of a few competing systems and projects for the next few years. Very, very few of us are likely to invest significant time in learning systems we do not already know or those that may be learned, at least superficially, in a few hours. This means that no system without a standard, GUI interface stands a chance of wide adoption. There seems to be a tacit, de facto agreement that the toolbox approach is best ... If I am right, then TUSTEP and CELLAR have the best chance of evolving toward the state at which our literary and linguistic colleagues are likely to pay attention to them. . . . In brief I intend it as encouragement to do what you are minded to do in any case: push the development of a Windows interface to TUSTEP as fast as you can.«

Damit bin ich bei dem Punkt, wo wir Sie einen ersten Blick auf unsere neueren Arbeiten werfen lassen und Sie zur Diskussion einladen möchten. Marc Küster wird eine Anwendung von TUSTEP für den Zugriff auf Daten vorstellen, bei der TUSTEP als Werkzeug hinter der Oberfläche gar nicht mehr sichtbar ist. Anschließend wird Matthias Kopp zeigen, wie wir TUSTEP auch als Werkzeug für die Erarbeitung von Daten attraktiver und leichter nutzbar machen möchten. (Wilhelm Ott)

#### TUSTEP als Werkzeug im WWW

Zu den großen Neuerungen des letzten Jahrzehnts gehört sicher das Web. Publikationen sollen gleichzeitig auf Papier und im Netz zu nutzen sein. Struk-

turierte Daten, z. B. Bibliographien, sollen bei Bedarf als Grundlage für anspruchsvolle Printpublikationen dienen, aber gleichzeitig über einen Web-basierten Datenbankserver online verfügbar gemacht werden können – ohne am Datenbestand selbst etwas ändern zu müssen.

Damit verbunden ist der Wunsch, nichtlateinische Texte in standardisierter, also für den Empfänger sofort zu entschlüsselnder Weise verteilen zu können – wiederum in ganz besonderer Weise dann, wenn Daten online verfügbar sein sollen. Hier bietet ISO/IEC 10646, auch bekannt als Unicode, eine Lösung. Eine Anwendung kann intern Daten in einer für den Verwendungszweck optimierten Weise erfassen und kodieren – der Weg des Language Tagging, den TUSTEP verwendet, ist einer der international anerkanntesten -, muß aber bei Bedarf transparent Daten in normierte Zeichensätze überführen können.

Mit der Internationalisierung à la Unicode geht paradoxerweise die Lokalisierung einher – zwei Seiten einer Medaille, die auch in der Industrie momentan ein ganz zentrales Thema ist. Nicht-englischsprachige Benutzer sind immer weniger gewillt, sich mit Lösungen abzufinden, die nicht ihren Bedürfnissen entsprechen. Sie erwarten zu Recht, die Schriftzeichen, die sie benötigen, auch verwenden zu können, die Sortierreihenfolge, die sie gewohnt sind und die den linguistischen Gegebenheiten ihrer Sprache angepaßt ist, auch vorzufinden und vieles mehr. Für die Geisteswissenschaften gilt dies in besonderer Weise, gibt es doch gerade in den Spezialdisziplinen viele für ihr Gebiet optimierte Konventionen, die »out of the box« von keiner Software unterstützt werden.

Alte und neue Anforderungen, von denen wir nur eine kleine Auswahl hier angerissen haben, kommen in vielen Varianten vor – differenziert durch spezielle Einsatzfelder, lokale Gegebenheiten, individuelle Vorlieben und vieles mehr. Dadurch daß kein einheitliches Anforderungsprofil existiert, kann es aber auch keine einheitliche Lösung geben. Für ernsthafte Anwender in Wissenschaft und Wirtschaft ist Flexibilität of the essence. Ein wirklich mächtiges Instrument muß programmierbar sein. Damit sind Flexibilität und leichte Benutzbarkeit zu einem gewissen Grad Widersprüche, die völlig zu überbrücken der Quadratur des Kreises gleichkommt. Wie Sie im Vortrag von Herrn Kopp hören werden, lassen wir uns trotzdem von dem Versuch der Quadratur nicht abschrecken – mit gutem Erfolg, wie ich meine.

Als Pilotprojekt, das aus meiner Arbeit als Leiter des Europäischen Projektteams »Browsing and Matching« hervorgegangen ist, haben wir einen OPAC für die Bibliothek des ZDV geschrieben. Die Anwendung demonstriert, wie einfach man mit TUSTEP im Hintergrund eine auf lokale Gegebenheiten optimierte Web-basierte Datenbankanwendung schreiben kann. Natürlich ist dieser OPAC nur ein Beispiel für die Möglichkeiten, mit TUSTEP-Makros – und mit TUSTEP-Bausteinen generell – anspruchsvolle Applikationen zu realisieren. Ganz andere Typen von Anwendungen – von maskenbasierten Eingabehilfen über Softwarelizenzierung und -download bis hin zu kompletten Redaktionssystemen – sind denkbar und teilweise bereits realisiert.

Damit ist es u. a. schnell und unkompliziert möglich, eine genau konfigurierbare intelligent unscharfe Suche (»intelligent fuzzy search«) zu programmieren.

Der OPAC berücksichtigt neben Phänomenen der deutschen Sprache (z. B. Gleichbehandlung von ä, a und ae; Gleichbehandlung von häufig in Namen vorkommenden Phonemen) auch z. B. die unterschiedliche Schreibung englischer und amerikanischer Begriffe: *centre* vs. *center*, *iz* vs. *is* (z. B. *realize*), *colour* vs. *color*. Weitere Anpassungen sind problemlos möglich.

Vollautomatisch und verlustfrei können die knapp 60.000 Datensätze der zugrundeliegenden Onlinedatenbank in die verschiedenen Speicherungsformate einschließlich Printversionen überführt werden. Für eine detailliertere Präsentation des OPAC möchte ich an dieser Stelle auf die Publikation in BI 98/9+10, S. 9-11 verweisen:

<http://www.uni-tuebingen.de/zdv/bilbi981bi98911-cgi.html>

und natürlich auf den OPAC selbst:

<http://www.uni-tuebingen.de/cgi-bin/zdvlit>

Die Makrosprache, in der der OPAC geschrieben ist, bietet dem Entwickler neben den üblichen Konstrukten einer modernen Programmiersprache mächtige, aber zugleich sehr flexible Spezialfunktionalität für die Textdatenverarbeitung. Zu den wichtigsten Features gehören die sog. X TABLEs, R-TABLEs und S-TABLEs und konsequent auf ihnen basierende Makrofunktionen. Die Tabellen gehören zu den Kernstücken der Makros und erlauben die bequeme Integration von TUSTEP in das CGI-Protokoll, auf dem (fast) alle interaktiven Anwendungen im Netz beruhen. Im einfachsten Fall sieht eine solche Anweisung wie folgt aus:

BUILD X TABLE Beispiel= "Ursprungsstring"Zielstring"

Durch diese Anweisung wird die Ursprungszeichenfolge in die Zielzeichenfolge umgewandelt oder zu Vergleichszwecken als zu der Zielzeichenfolge äquivalent behandelt. Natürlich sind die Einträge frei wählbar; genaue Suchbedingungen und Ausnahmen können in der in TUSTEP üblichen Syntax für regular expressions – deren Mächtigkeit über die in der UNIX-Welt verbreiteten weit hinausgehen – angegeben werden. Es steht die volle Mächtigkeit der aus der zeige-Anweisung des Editors und aus den entsprechenden Parametern von #KOPIERE bekannten Zeichenfolgen-Such-Tabellen zur Verfügung.

Mit den STRUCTURE-Anweisungen, einer weiteren Neuerung der Makro-Umgebung, ist es leicht möglich, eine Anwendung auf Daten mit fast beliebiger Struktur aufzusetzen. Die STRUCTUREs reflektieren direkt die Struktur (oder die Strukturen) der zugrunde liegenden Daten. So hat die dem OPAC konkret zugrunde liegende Datenbank zwei konkurrierende Strukturen für selbständige und unselbständige Publikationen. Es ist hier problemlos möglich, auch komplexe SGML-Strukturen zu verwenden.

Gemäß der TUSTEP-Philosophie kann (und muß) auch die Textausgabe völlig frei gestaltet werden können. Über den Tabellenmechanismus kann man allerdings die TUSTEP-interne Kodierung bequem in eine große Auswahl von üblichen Codierungsschemata einschließlich Unicode umsetzen.

Nicht unwesentlich unterscheidet auch sein durchdachtes Sicherheitskonzept TUSTEP von anderen Produkten. Ein Makro – und damit erst recht der Endnutzer – kann nur auf explizit freigegebene Ressourcen auf dem Server zugreifen. So müssen z. B. Dateien vor der Benutzung angemeldet werden und andere Ordner über sog. Trägervariablen, die nicht zur Runtime manipuliert werden können, bewußt sichtbar gemacht werden. Nach menschlichem Ermessen ist es damit für einen potentiellen Eindringling unmöglich, aus dieser Sandbox auszubrechen und unerlaubte Informationen jedweder Art zu sammeln.

Die Einbindung von TUSTEP über CGI ermöglicht es jetzt, die Mächtigkeit und die Zuverlässigkeit von TUSTEP zu nutzen, ohne von Nutzern zu verlangen, auf ihre gewohnten Oberflächen – ihren Browser in diesem Fall – zu verzichten oder das Wort TUSTEP auch nur je gehört zu haben. Somit kann man TUSTEP auch als eine hochoptimierte und doch überaus flexible Programmierungsumgebung zur Textdatenverarbeitung verstehen, mit deren Makrosprache man in kurzer Zeit und mit geringem Aufwand komplexe Applikationen erstellen kann. Die Anzahl der möglichen Anwendungsfelder wird dabei praktisch nur durch die Phantasie des Programmierers beschränkt.

*(Marc Wilhelm Küster)*

#### Interaktive Oberfläche

Mit der interaktiven Oberfläche, die seit einigen Monaten in der Abteilung LDDV entwickelt wird, soll vor allem Neueinsteigern der Zugang zu TUSTEP erleichtert werden.

Um dieses Ziel zu erreichen, werden – alternativ zur bisherigen Befehlsnotation – Menüstrukturen und graphische Steuerelemente angeboten. Ein Teil der Leistungen von TUSTEP wird damit erstmals auch unter einer graphischen Benutzeroberfläche verfügbar.

Diese Veränderung der Oberfläche hat wesentliche Konsequenzen: Die Aktivitäten des Benutzers verändern sich nicht nur in mechanischer Hinsicht. Mit dem Wechsel von der Kommandoeingabe zur Bedienung von Buttons und Menüs ergibt sich die Möglichkeit zu konzeptionellen Veränderungen: Neben die bisherigen TUSTEP-Module, die nicht nur technische Kenntnisse, sondern auch fachwissenschaftliche Kenntnisse und vor allem fachwissenschaftliches Problembewußtsein erfordern, treten nun anwendungsnähere Module, die einfach bedienbar sind und – ohne die erwähnten Voraussetzungen – standardisierte Lösungen anbieten. Neben das Werkzeug, das in eigener Verantwortung eingesetzt, konfiguriert und speziellen Problemstellungen angepaßt werden kann, tritt die vereinfachte Lösung, die bereits nach wenigen einfachen Entscheidungen ein Resultat liefert.



Demonstriert wurde ein Modul für die Registererstellung, das prototypisch die Möglichkeiten der neuen Oberfläche erkennen läßt. Für die interaktiv gesteuerte Erstellung von Registern stellt es prinzipiell zwei Wege zur Verfügung.

Für Anfänger steht ein Modus zur Verfügung, bei dem der Anwender Schritt für Schritt geführt wird und jeweils nur eine Entscheidung zu treffen hat:

- Zunächst wird die Quelldatei festgelegt. Falls ihr Name nicht bekannt ist, kann die Verzeichnishierarchie des jeweiligen Arbeitsplatzes mithilfe graphischer Steuerelemente durchsucht werden. Das Resultat wird durch Mausklick bzw. Tastendruck übernommen.
- Im nächsten Schritt wird der Typus des Registers festgelegt. Zur Auswahl stehen momentan *Wortformenliste*, *KWIC-Index* sowie ein auf speziell markierten Termini basierendes »klassisches« *Register*; im letztgenannten Fall werden die erforderlichen Kennzeichen erfragt und auf Konsistenz überprüft.
- Nachdem entschieden wurde, ob Referenz- und Häufigkeitsangaben verwendet werden sollen, wird die Sortierrichtlinie festgelegt, d. h. die Sprache wird angegeben, deren Sortierregeln gelten sollen.

Eine synoptische Darstellung aller getroffenen Entscheidungen schließt den Auswahlprozeß ab. Alle Entscheidungen können an dieser Stelle noch einmal modifiziert werden. Eine Hilfefunktion stellt jeweils Informationen zur aktuellen Entscheidung zur Verfügung. Alternativ zum geschilderten schrittweisen Vorgehen kann die synoptische Darstellung auch sofort erreicht werden; dieser Modus richtet sich an fortgeschrittene Benutzer, die erste Erfahrungen mit dem Modul gesammelt haben. Nach Ausführung des zusammengestellten »Programmes« kann das Resultat am Bildschirm betrachtet oder ausgedruckt, natürlich auch gespeichert werden.

Weiterhin besteht die Möglichkeit, das Programm zu modifizieren, erneut auszuführen und unter einem benutzerdefinierten Namen zu speichern. Unter diesem Namen kann es ggf. zu einem späteren Zeitpunkt wieder aufgerufen, ausgeführt, modifiziert werden. Nach wie vor bewegt sich der Benutzer dabei auf einer anwendungsnahen Ebene. Er entscheidet sich z. B. für oder gegen absolute und relative Häufigkeitsangaben, benötigt also keinerlei Wissen über die TUSTEP-Module, deren Spezifikationen und Parameter. Intern werden die ein Register determinierenden Benutzereingaben in einem Metaformat verwaltet und gespeichert. Aus diesem Metaformat wird bei der Ausführung ein »traditionelles« TUSTEP-Programm erzeugt. Dieses Programm enthält umfangreiche Kommentare. Es kann betrachtet, ausgedruckt und ggf. in die Kommando-Umgebung exportiert werden. Die Kommentare erläutern die programmtechnischen Konsequenzen unterschiedlicher Entscheidungen.

Die interaktive Oberfläche führt den Benutzer gleichermaßen an fachwissenschaftliche wie an programmtechnische Probleme heran. Dies wurde bereits angedeutet, sei nun am Beispiel der Sortierrichtlinien verdeutlicht: In der Regel

wird ein Anfänger nicht wissen, daß es im Deutschen für die Behandlung der Umlaute zwei unterschiedliche Normen des DIN gibt; ebenso wenig wird er imstande sein, die in TUSTEP für diese und für andere, benutzereigene Definitionen der Sortierfolge vorgesehenen Mechanismen einzusetzen. Unter der interaktiven Oberfläche wird zunächst eine Auswahlmöglichkeit angeboten, die durch einen Hilfetext erläutert wird. In einem zweiten Schritt kann sich der Benutzer anhand des kommentierten Programmcodes über die programmtechnischen Details informieren, die zur Umsetzung derartiger Entscheidungen eingesetzt werden können. In einem dritten Schritt schließlich kann er, nun nicht mehr unter Verwendung der interaktiven Oberfläche, sondern in der traditionellen TUSTEP-Umgebung, ein auf dieser Grundlage erstelltes und exportiertes Programm verfeinern und in der »traditionellen« Kommando-Umgebung verwenden.

Abschließend seien vier wesentliche Aspekte festgehalten:

- 1) Abstraktion: Zwischen dem Anwender und seinen Problemen auf der einen, den Daten und Programmen auf der anderen Seite wird eine Zwischenebene eingezogen, auf der die Anwendungsprobleme abstrakt beschrieben werden. Sie können damit reproduzierbar in Programmcode übertragen werden.
- 2) Anwendungsnahe: Verallgemeinerte und vereinfachte Lösungen werden angeboten, der Benutzer benötigt keine speziellen Kenntnisse über z. B. Sortierschlüssel, spezifische Sortierregeln usw.; stattdessen trifft er auf einem anwendungsnahen Niveau Auswahlentscheidungen
- 3) Leistungsumfang: Unter der demonstrierten interaktiven Oberfläche ist nicht das volle Leistungs- und Differenzierungsvermögen von TUSTEP verfügbar. Dies ist, abgesehen von der Programmierungsökonomie, durch die Zielsetzung bestimmt: Eine Vielfalt von Möglichkeiten wirkt – unabhängig davon, ob sie qua Kommandooption oder unter zahlreichen Menüs und Submenüs verfügbar gemacht werden – auf Anfänger verwirrend.
- 4) Didaktik: Die interaktive Oberfläche soll den Benutzer zum Kommandomodus hinführen. Aus diesem Grunde ist es möglich, erstellte Programme nicht nur in der Sicht der interaktiven Oberfläche als eine Sammlung diverser Auswahlentscheidungen zu betrachten, zu modifizieren, zu speichern; ebenso kann auch der generierte (und kommentierte) Programmcode betrachtet, ausgedruckt, exportiert, modifiziert und erweitert werden.

*(Matthias Kopp)*